ECE 3663 - Design Project Final Report

Roman Boyarov
Apollinaire Mase
Adam Abdulhamid
ECE 3663 – Spring 2013
University of Virginia

Abstract

This project consists of an embedded Digital Signal Processor design using Cadence and the FreePDK package. This design is a proposal for Itty Bitty Logic Co. which consists of building two registers and an Arithmetic Logic Unit (ALU) containing several functions. The ALU consists of MUXes, an XOR gate, an Adder, Two's complement, Mux, AND gate and so on…As an approach for this design innovative solutions were used to implement tradeoffs that minimized the relative size, delay, and power of the project. On a basic level, traditional PMOS and NMOS transistors were used to realize gates and other functions. For consultation we used resources from Rabaey's book, Professor Blalock, and TA Alicia Klinefelter.

1. Introduction

Itty Bitty Logic Co., a leader in developing embedded Digital Signal Processor, has launched a program that gives startup companies an opportunity to demonstrate their talents. In order to be considered as a potential contractor, a design project has been organized by Prof. Blalock, a liaison agent from the group IBL. The aim of this project is to demonstrate our capability to win the best design. Our design implements many common functions of an ALU, as required, using the software Cadence and the freePDK package, with some innovations and optimizations including sizing and delay. The present report presents all about the techniques used in order to win the contest. Given the information in this proposal, we hope that our design will be selected for development.

2. Design description

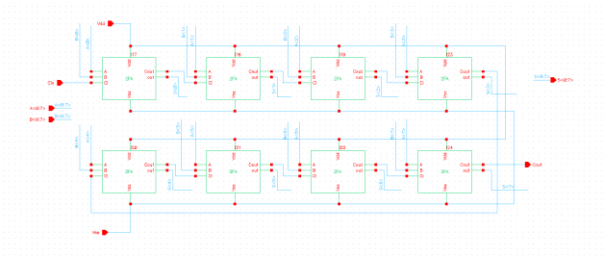
On a level of abstraction, the design consisted of two registers, and an ALU. The input of the design feeds through the first register (clocked) to one input of the ALU. The other input of the ALU comes from the output of the system. The ALU is controlled by a 3bit Control input. The output of the ALU then feeds through the second register (clocked).

The registers were D-Flip Flops – which is just a negative edge triggered flip flop connected to a positive edge triggered flip flop (in that order).

The ALU consists of a range of functions that are fed through a MUX. The MUX chooses what is outputted through its 3bit Control input. The clock is controlled by an input at a certain frequency.

2.1 The Adder. A simple approach

The adder has been chosen from Rabaey's book on page 567. The schematic of a two full bit mirror adder realized in contained inside the picture files. The 8bit version consists of putting together 8 2-bit adders. It is possible to create a logical circuit using multiple full adders to add N-bit numbers. After conducting numerous tests on the two inputs adder, we observed that the results obtained followed the expected values from a simple truth table. It has been chosen because it requires only 24 transistors. It presents a considerable reduction in both delay and area than the full adder.

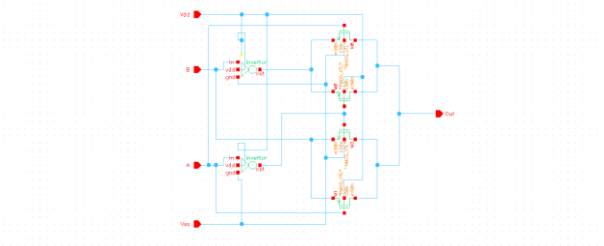


2.2 Two's Complement.

It consists of passing and input through an inverter then simply adding 1 to it. The function of the two's complement was made possible through the use of our previous schematics (adder and inverter). For a 8-bit two's complement, instead of having 8 extra adders on top of the 8 last ones, the multiplexer helped to manage its integration, which considerably helped to reduce the size of the function 8-bit two complement.

2.3 The XOR gate. An Efficient design

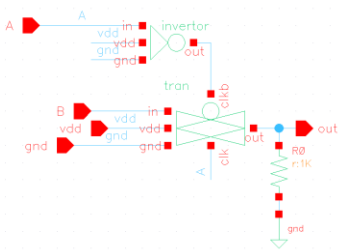
The XOR was built from the idea of minimizing the size since the previous 8-bit adder occupies a non negligible area. Made out of transmission gate, it only takes 4 transistors considering that the complements of the two inputs are provided.



2.4 Basic Blocks-AND and OR

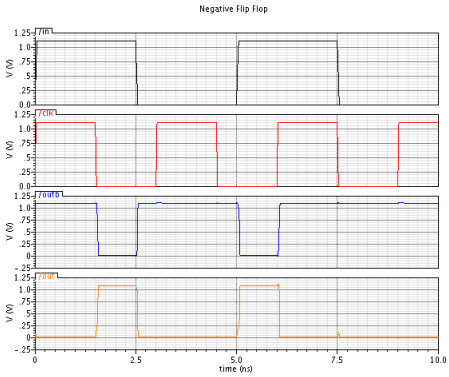
For the AND and OR blocks, we played around with some different configurations. The basic one was implemented using the pull up and pull down network. This one guarantees a good

swing at the output and doesn't create any inputs that would create a conflict at the output. We also created this using pass gate logic, as the XOR was implemented. While this does reduce the area somewhat, considering we still need to use an invertor at the input to have all the inputs we need, it was not a significant decrease. We also ran the simulations to see the comparison for power and delay, and these too were not improved significantly, and so we decided to stay with our first design using the pull up and pull down networks. The schematic for the AND gate created using pass gates can be seen here:

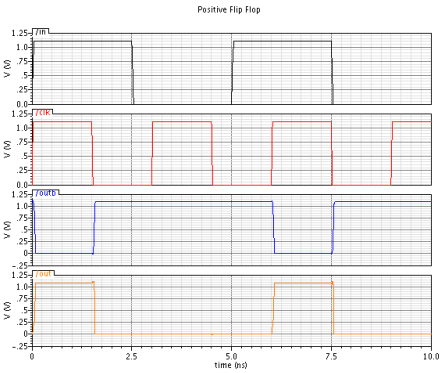


2.5 Register File

The registers are the system for holding the values sent to them. The one's we used were based off negative edge triggered flip flops. Looking at the design for one bit, because the 8-bit register file is simply multiple of these, we have a master-slave flip flop system in place. The master flip flop sends the value through on the bottom half of the clock cycle, and the rest is sent through on the upper bound of the clock cycle, thus creating a full change from input to output over one clock cycle. In our simulations, we tested each of the separate flip flops first, and they worked perfectly



.



They each passed the value through on their respective transparency phases. However, when we constructed the master-slave flip flop for the register, we ran into some problems where the output would follow the edge of the clock when the input was low. We believe this to be due to induced capacitance between the flip flops.

2.6 Transistor Sizing

When it came to sizing, we wanted to minimize delay as much as possible, even if at a cost for area because there was a stronger weight on the delay than area using our final metric of area*delay^2*power. We first needed to find the worst case delay to know where to put the most effort in optimizing for delay. While the adder is the worst-case single block, the two's compliment function ends up being the absolute worst-case because it uses the invertor block followed by the adder block. The invertor block was already optimized as much as possible with respect to the minimum of the NMOS gate, being 50nm by 90nm. We sized the all the gates in the adder to have the same characteristic delay paths as that of the invertor. By now increasing the sizes of the adder to minimize on the worst case delay, we could now actually increase the delay of the others and save area as long as they never exceeded the current worst case delay. This is what will be used as the determining factor for the max clock speed. However, any block can only be minimized to the size of the basic NMOS transistor, because that is a limiting factor by fabrication process, one that we have no control over.

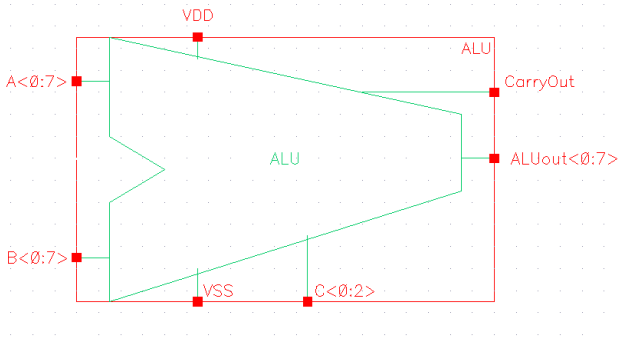
Table 1: Sizing (NMOS = 50nm *90nm)

Component	Size (multiple of NMOS)
OR	80
AND	64
XOR	48
Adder	776
2 bit Adder	36
Twos Comp	800
Total	**1804**

3. ALU Construction

The ALU provides 8 different functions that can be run, determined by a three bit input control. Our team took into account several things in the overall design of how the ALU was laid out. The majority of the functions were independent blocks, such as the OR and AND functions. When it came to the two's compliment, it utilizes an invertor block followed by an adder. To save on total power and size, we used the adder we already have as a function block rather than having a whole separate one. To accomplish this, we needed to have a two to on mux that was controlled by the bit that changed between the adder and two's compliment on the control line for the entire ALU. When the two's compliment was called, it would read in the output from the invertor block, and add it with a stationary byte that represented a one. We also placed two 8 to one muxs on the inputs and one at the output. By doing this, only one function block is running and the outputs are not calculated and wasted. This would have increased the overall power draw because regardless of which function is called, all would have run. By the placement of the

various muxs, we made an innovation over the basic system and taking the compromise to add some to the overall area of the system, by using the muxs, but reducing the overall power.



.

4. Power

The power for each block was found through cadence simulation. We used an arbitrary resistor at the output for each function block, and controlled the voltage of the input. In this sense, we would black box the function block itself, and look at everything else in a series circuit we would create and deduce what we needed about the function block. Through simulation, we could calculate the voltage across the resistor we put in, and through this, we could calculate both the voltage across the function block, as well as the current. We made sure to take the average values over multiple clock cycles for accurate values. Now we had everything for power calculations (using the most basic power=current * voltage equation).

Table 2: Power Dissipation

Component	Power (microW)
OR	296
AND	156
XOR	147.2
Adder	289
Twos Comp	326
Total	**1214.2**

5. Delays

Table 3: Time Delay

Component	Delay (ps)
OR	10.5
AND	14.28
XOR	3.08
Full Adder	802
2 bit Adder	111
Twos Comp	804
Worst Case	**804**

6. Conclusion

After integrating numerous functions and logical tradeoffs to attempt to get this project to realize the proposed functionality, we understood the purpose of IBL's contest. Along with giving students the power to do work for a company it gave us a feel for how such a working environment can be and the many pitfalls and hardships you will face. We obtained functionality for all functions and gates. We believe all our topology to also be correct. The final design metric was found to be (1804*90nm)*1214.2microWatts*804ps. An important part of this project was to be able to generating a test bench using numerous tools (Ocean, specter...), that each has their benefits. We are confident about our approaches but understand that our design fell slightly short of the full design specifics. We would be eager to improve on our results in future work. All schematics can be found in the external document titled 'Design Project Figures.'

7. Acknowledgements

We gained a lot of experience from this project and would like to thanks the University of Virginia for providing all the tools for this project. A special thanks is addressed to Prof. Blalock and Alicia Klinefelter for their support in this learning process.

8. References.

Digital integrated Circuits. JAN M. RABAEY
Cadence Software on Nx